

EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.
2. Authorization for this examiner's amendment was given in a telephone interview with Yefin Zhuk on May 19, 2010.
3. Claims 4-6, 8, and 10-16 (renumbered as 1-11) has been allowed.
4. The application has been amended as follows:

Cancel claims 1-3, 7, and 9.

REPLACE Claims 4-6, 8, and 10-16 with claims 4-6, 8, and 10-16 amended by examiner (without underlined and cross marked) set forth below:

Claim 4. Knowledge-driven architecture control system that combines hardware and software components and services including audio and video systems, input transformation components and distributed networks, comprising:

- (i) Knowledgebase comprising semantic-enabled rules engine component, containing business domain ontology and business rules, and application scenarios that reflect application requirements;
- (ii) ApplicationScenario Player capable of transforming acts of scenarios and business rules into interactions with knowledgebase, presentation components, and the underlying application services;
- (iii) ServiceConnector transforms service requests from application scenario acts into direct calls to service components, wherein the Service connector

comprising (a) Object Retrieval that is able to find an existing service object or load the requested service class and instantiate the object at run-time (b) Object Registry that associates service objects with service and object names, stores service objects, and makes them reusable (c) Method Retrieval that retrieves the proper service method belonging to a selected service object based on the provided method arguments (d) Method Performer that performs the requested service operation on the selected service object;

(iv) Service components comprising integration-ready components with separate APIs required by the ServiceConnector;

(v) Presenter comprising (a) Formatter that prepares data for audio or video interaction or for communication to other programs and passes data further to (b) Performer that uses formatted data for actual presentation to one or more agents via voice or screen or electronic formats for different types of agent devices; wherein the knowledge-driven architecture control system comprising:

the ScenarioPlayer receives an XML instruction from the network, as an act of a current scenario, or a user's input related to the scenario, analyses successful scenario execution including: a) history of successes b) history of interpretation failures c) learning scenarios that prompt an agent, a user, or a program to redefine the input, or to provide more details for better interpretation d) queue of scenarios with un-answered questions to resolve unsuccessful interpretations, upon successful execution, the ScenarioPlayer accesses the Knowledgebase to interpret the input and translate said input into a service request directed to the

ServiceConnector; the ServiceConnector accepts service and action names as parameters and connects to or obtains a necessary service object that will perform the requested operation/method, wherein the service object invokes required method and parameters, and delivers results back to the ScenarioPlayer; the ScenarioPlayer gets the results of the service object, and passes the results to the Formatter wherein the Formatter translates the results into a presentation format and produces XML scenarios related to the expected user interaction; and the ScenarioPlayer interprets the results for the Performer object, wherein the Performer object presents results on a screen or/and in a voice format.

Claim 5. The Knowledge-driven architecture control system of claim 4, further comprising: input transformation hardware and software components, comprising optical and mechanical speech, handwriting, and image recognition components; wherein said input transformation components are connected and interact to the knowledgebase component filled with expected patterns and recognition scenarios and rules which provide transformation of multiple input types into traditional text and numeric variable values expected by event handling scenarios.

Claim 6. The Knowledge-driven architecture control system of claim 4, wherein the knowledgebase component further comprising a service adapter to the knowledgebase, providing a standard service interface

required by the ServiceConnector, which interacts with the knowledgebase as with a set of services and a knowledge engine.

Claim 8. The Knowledge-driven architecture control system of claim 4, further comprising the Optimizer component wherein the Optimizer takes a snapshot of existing rules and scenarios and translates them into a source code comprising java and C## languages, which can later be compiled into binary code to fix the current application rules into a regular application.

Claim 10. The Knowledge-driven architecture control system of claim 4, wherein service components are endowed with usage and value properties.

Claim 11. The Knowledge-driven architecture control system of claim 10, wherein the Success Analysis component maintains and consistently refines a list of previously used services with their APIs, keywords, descriptions, and related scenarios in the knowledgebase, and re-evaluates the usage and value properties of the services.

Claim 12. The Knowledge-driven architecture control system of claim 4, wherein the New Agent Request interpreter uses the list of previously used services with their APIs, keywords, descriptions, and related

scenarios for automatic translation of user requests into service APIs and scenario acts.

Claim 13. The Knowledge-driven architecture control system of claim 4, wherein the New Agent Request interpreter uses a list of previously used services with their APIs, keywords, descriptions, and related scenarios to offer selected parts of this information to the user for semi-automatic translation of new requests into service APIs and scenario acts.

Claim 14. The Knowledge-driven architecture control system of claim 4, further comprising a Communicator that provides collaborative access to knowledge and services existing on other distributed network systems built with this architecture.

Claim 15. The Knowledge-driven architecture control system of claim 11, wherein the Success Analysis component propagates via the Communicator to distributed network systems information on a new service API or a new knowledge subject after the first success operation that included the service or the knowledge subject and then after each update provided locally by the Success Analysis component.

Claim 16. The Knowledge-driven architecture control

system of claim 15, wherein information on new elements is propagated after the first successful operation that included the new element and thereafter after each local update of such information by the Success Analysis component.

The following is an examiner's statement of reasons for allowance:

The prior art of record fails to teach or suggest the claimed invention individually or in combination the limitation of "the ScenarioPlayer receives an XML instruction from the network, as an act of a current scenario, or a user's input related to the scenario, analyses successful scenario execution including: a) history of successes b) history of interpretation failures c) learning scenarios that prompt an agent, a user, or a program to redefine the input, or to provide more details for better interpretation d) queue of scenarios with un-answered questions to resolve unsuccessful interpretations, upon successful execution, the ScenarioPlayer accesses the Knowledgebase to interpret the input and translate said input into a service request directed to the ServiceConnector; the ServiceConnector accepts service and action names as parameters and connects to or obtain a necessary service object that will perform the requested operation/method, wherein the service object invokes required method and parameters, and delivers results back to the ScenarioPlayer; the ScenarioPlayer gets the results of the service object, and passes the results to the Formatter wherein the Formatter translates the results into a presentation format and produces XML scenarios related to the expected user interaction; and the ScenarioPlayer interprets the results for the Performer object, wherein the Performer object presents results on a screen or/and in a voice format" as set forth in claim 4.

The closest newly cited prior art, Jasper et al. discloses a framework for understanding and classifying ontology applications. Leff et al. discloses a Web-Application development using the Model/View/Controller design pattern. Clark discloses a ruled-based method for testing of programming segments. Bailer et al. discloses a method for analysis of software requirements. Green et al. discloses a method for an n-tier software component architecture application. However, the prior art does not teach or suggest the limitations cited above as being free of any prior art when read in the claims as a whole.

CONCLUSION

5. Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

6. Patent applicants with problems or questions regarding electronic images that can be viewed in the Patent Application Information Retrieval system (PAIR) can now contact the USPTO's Patent Electronic Business Center (Patent EBC) for assistance. Representatives are available to answer your questions daily from 6 am to midnight (EST). The toll free number is (866) 217-9197. When calling please have your application serial or patent number, the type of document you are having an image problem with, the number of pages and the specific nature of the problem. The Patent Electronic Business Center will notify applicants of the resolution of the problem within 5-7 business days. Applicants can also check PAIR to confirm that the problem has been corrected. The USPTO's Patent Electronic Business Center is a complete

service center supporting all patent business on the Internet. The USPTO's PAIR system provides Internet-based access to patent application status and history information. It also enables applicants to view the scanned images of their own application file folder(s) as well as general patent information available to the public.

7. For all other customer support, please call the USPTO Call Center (UCC) at 800-786-9199. The USPTO's official fax number is 571-272-8300.

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to C. Dune Ly, whose telephone number is (571) 272-0716. The examiner can normally be reached on Monday-Friday from 8 A.M. to 4 P.M.

9. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim Vo, can be reached on (571) 272-3642.

/Cheyne D Ly/
Primary Examiner, Art Unit 2168